Education and Information Technologies
https://doi.org/10.1007/s10639-025-13452-5

MANUEL B. GARCIA
www.manuelgarcia.info

Journal Article

Teaching and Learning Computer Programming Using ChatGPT: A Rapid Review of Literature Amid the Rise of Generative AI Technologies

Manuel B. Garcia [a,b,c] *

[a] College of Education, University of the Philippines Diliman, Quezon City, Philippines
[b] Educational Innovation and Technology Hub, FEU Institute of Technology, Manila, Philippines
[b] College of Education, Korea University, Seoul, South Korea

*** Correspondence:**
Manuel B. Garcia, University of the Philippines Diliman and FEU Institute of Technology.
mbgarcia@feutech.edu.ph

How to cite this article:
Garcia, M. B. (2025). Teaching and Learning Computer Programming Using ChatGPT: A Rapid Review of Literature Amid the Rise of Generative AI Technologies. *Education and Information Technologies*. https://doi.org/10.1007/s10639-025-13452-5.

Article History:
Received: 1 Mar 2024
Revised: 9 Jan 2025
Accepted: 17 Feb 2025
Published: 1 Mar 2025

Abstract:

The emergence of generative AI tools like ChatGPT has sparked investigations into their applications in teaching and learning. In computer programming education, efforts are underway to explore how this tool can enhance instructional practices. Despite the growing literature, there is a lack of synthesis on its use in this field. This rapid review addresses this gap by examining the current literature to outline research trends, assess how it supports teaching and learning processes, and discern the issues that emerge from its application in programming instruction. A total of 107 documents disseminated across 81 distinct sources and authored by 394 contributors were identified. The review adopted a broad and inclusive approach, selecting literature based on relevance to ChatGPT's application in programming education and encompassing studies from diverse settings and methodologies. Results highlight applications such as personalized tutoring, knowledge reinforcement, instructional material creation, source code generation, immediate feedback, and assessment support. However, its use also introduces challenges such as academic dishonesty, ethical dilemmas, diminished critical thinking, overdependence on ChatGPT, and various technical limitations. Considering these findings, a balanced approach to the utilization of ChatGPT in programming education is essential. Implications and recommendations have been provided to guide policymakers, curriculum designers, teachers, and students in harnessing the benefits of this technology while mitigating potential challenges.

Keywords:

Artificial Intelligence, Generative AI, ChatGPT, Large Language Model, Computer Programming, Coding, Literature Review, Rapid Review

This is a pre-copyedit version of an article copied from https://manuelgarcia.info/publication/chatgpt-programming-education and published in the *Education and Information Technologies*. The final authenticated version is available online at https://doi.org/10.1007/s10639-025-13452-5. Any other type of reproduction or distribution of the article is not authorized without written permission from the author and publisher.

INTRODUCTION

Computer programming is a vital area of research. The digitization of virtually all aspects of human life has underscored its indispensable role in modern society. According to the Asian Development Bank (2022), programming skills are progressively becoming a requisite across various professional fields worldwide. This shift reflects a fundamental transformation in the skill sets required for professional success and societal contribution (Tuomi et al., 2018). Notably, this trend is not confined to the technology sector alone. Fields as varied as healthcare, finance, education, agriculture, and the arts, where coding proficiency was once merely beneficial (Guo, 2013), now require it as an essential skill. The International Labour Organization (2021) corroborated this perspective, revealing that coding skills are now required in many professions (e.g., engineers, scientists, and artists). As the global economy grows more dependent on digital technologies, this shift underscores the importance of programming skills not only for personal career growth but also for enhancing national competitiveness and fostering economic development. Countries that invest in expanding their digital workforce and integrating programming education at all levels are positioning themselves to lead in the next wave of technological advancements (OECD, 2019). Recognizing this growing importance, various countries are launching national initiatives aimed at enhancing their programming education infrastructure, from grassroots coding programs for children to advanced professional retraining for adults (Asian Development Bank, 2022). Consequently, computer programming has emerged as a critical area of interest in educational research, with pressing questions about how best to teach and integrate programming skills into curricula at all levels of education.

In scientific literature, teaching and learning programming has been a common subject of many investigations (Scherer et al., 2020). These studies often focus on identifying the most effective instructional approaches and assessing their effectiveness in addressing learning difficulties. Unfortunately, the educational process in programming continues to be a significant challenge, as demonstrated by high dropout and failure rates (Garcia, 2023a, 2024). This situation has spurred ongoing research efforts to find more effective instructional tools and educational approaches. Recently, the advent of ChatGPT has prompted investigations into its possible application in programming education (Deriba et al., 2024; Husain, 2024). These explorations are motivated by ChatGPT's potential as a virtual tutor and learning companion that can provide personalized guidance and support to students. Although other large language models (LLMs) for programming exist (e.g., GitHub Copilot, Amazon CodeWhisperer, Meta Code Llama, and OpenAI Codex), ChatGPT was selected because many studies have already provided initial insights into the benefits and challenges of using it as an educational tool in programming education (Bringula, 2024). However, there remains a significant research gap in synthesizing the extent of its integration, utilization, and effectiveness within this domain. Therefore, this study aims to fill this gap by conducting a rapid literature review that answers the following research questions (RQs):

RQ1. What are the trending research topics related to ChatGPT in programming?

RQ2. How does ChatGPT assist in teaching and learning programming?

RQ3. What issues arise from using ChatGPT in programming instruction?

METHODOLOGY

Research Design

ChatGPT is an emerging technology in the education sector (Baig & Yadegaridehkordi, 2024). Despite the availability of other generative artificial intelligence (AI) tools specifically designed for coding, such as *GitHub Copilot*, *Replit GhostWriter*, and *Amazon CodeWhisperer*, ChatGPT stands out as the most widely adopted and versatile tool for both general and educational purposes. With increasing interest from the academic community, there is a pressing need to understand its applications and implications. Despite the growing curiosity, the literature on its educational usage is still developing. This limitation means conducting a comprehensive systematic review could be a potentially lengthy process with limited existing studies to analyze (Khangura et al., 2012). Given these considerations, the methodology of this chapter pivots towards a rapid review approach. A rapid review is a streamlined method for synthesizing research in a time-efficient manner (Tricco et al., 2015), ideally suited for fields where the body of literature is either emerging or in the process of rapid expansion. This approach is particularly advantageous when timely insights are necessary to inform practice or policy or when resources are constrained. Opting for a rapid review in this context allows us to gather, assess, and integrate the available research on the role of ChatGPT in programming education. Additionally, the paper incorporated a bibliometric analysis to examine the current state of the literature. This approach was integrated to provide a quantitative overview of the research trends, publication patterns, and thematic focuses within this field (Garcia, 2023b). This combined methodology is deemed the most appropriate given the current scope of literature and the need for a swift yet thorough understanding of its applications and potential challenges in programming instruction.

Search Strategy

This rapid review adhered to the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) guidelines to ensure a structured and transparent approach to selecting relevant articles. The initial literature search was conducted on 9 February 2024, using the Scopus database as the primary source for academic publications. To broaden the scope, the Web of Science (WoS) database was included in a second search on 16 September 2024. A third search was conducted on 9 January 2025, incorporating IEEE Xplore, to capture recent studies not yet indexed in Scopus or WoS. These additional searches were undertaken proactively, following the receipt of reviewers' feedback, to ensure the review remained updated and comprehensive. The search strategy utilized the following query: (*programming* OR *coding* OR "*computer programming*") AND (*education* OR *teaching* OR *learning* OR *instruction* OR *pedagogy* OR *curriculum* OR *e-learning* OR "*online learning*" OR "*distance learning*") AND *ChatGPT*. This approach targeted publications that included these terms in their titles, abstracts, and keywords. It was designed to capture a comprehensive set of studies focusing on the use of ChatGPT within the context of programming education across various educational settings and modalities. The search did not limit the period of publication, assuming literature on ChatGPT remains sparse (Stone, 2023). The selection process focused exclusively on journal articles, conference papers, and book chapters written in English. Studies were excluded if they lacked empirical data, such as conceptual papers, opinion pieces, or editorials. Papers that discussed AI broadly without a

specific focus on ChatGPT were also excluded to maintain relevance. Other types of documents, duplicate records, and studies not directly related to the topic were omitted to refine the analysis. To avoid bias and ensure a balanced exploration of both the benefits (RQ2) and issues (RQ3) associated with the use of ChatGPT in programming education, all publications that met the selection criteria, whether positive or critical, were included. Additionally, two external evaluators—both experienced programming professors—were recruited to independently review and verify the selected studies, further ensuring the neutrality and rigor of the review process. Details of the databases, search dates, and rationales are summarized in Table 1.

Table 1. Search Strategy and Database Rationale

Database	Search Date(s)	Rationale
Scopus	9 February 2024 16 September 2024 9 January 2025	Primary source for academic publications due to its comprehensive coverage of peer-reviewed literature.
Web of Science	16 September 2024 9 January 2025	Added to broaden the scope and include multidisciplinary sources not indexed in Scopus.
IEEE Xplore	9 January 2025	Included to capture engineering and technology-focused studies not yet indexed in other databases.

Data Analysis

All relevant publications, along with their metadata, were saved and downloaded in a *.csv* format. Firstly, the file was imported into *Posit Cloud* (formerly known as *RStudio*) to conduct bibliometric analysis, which is a method used to assess research trends and patterns by examining publication data. The trending research topics related to ChatGPT in programming were identified using the *bibliometrix* package (RQ1). This open-source package is designed to conduct quantitative analysis in the fields of bibliometrics and scientometrics (Aria & Cuccurullo, 2017). The bibliometric analysis approach was guided by the technique toolbox proposed by Donthu et al. (2021). Subsequently, the data extracted were then processed through three principal coding phases (Creswell, 2012). This analytical process was undertaken to explore how ChatGPT is utilized in the context of teaching and learning (RQ2). The first phase, known as open coding, involves the initial examination of the data, where codes are developed to describe and categorize the information, facilitating a profound exploration without the influence of pre-existing notions. Following this phase, axial coding is applied to examine the connections and relationships among the codes identified during the open coding phase. This step categorizes the codes into broader themes or categories, delineating the relationships among them. The final phase, selective coding, further refines these categories or themes, aiming to construct a cohesive narrative. This phase focuses on identifying the core category or the primary phenomenon that the data reveals. Within this analytical framework, issues and challenges associated with the use of ChatGPT in programming instruction were also expected to surface (RQ3). These concepts would be discerned as distinct themes during the coding process, providing insights into potential obstacles and considerations for integrating ChatGPT into programming curricula.

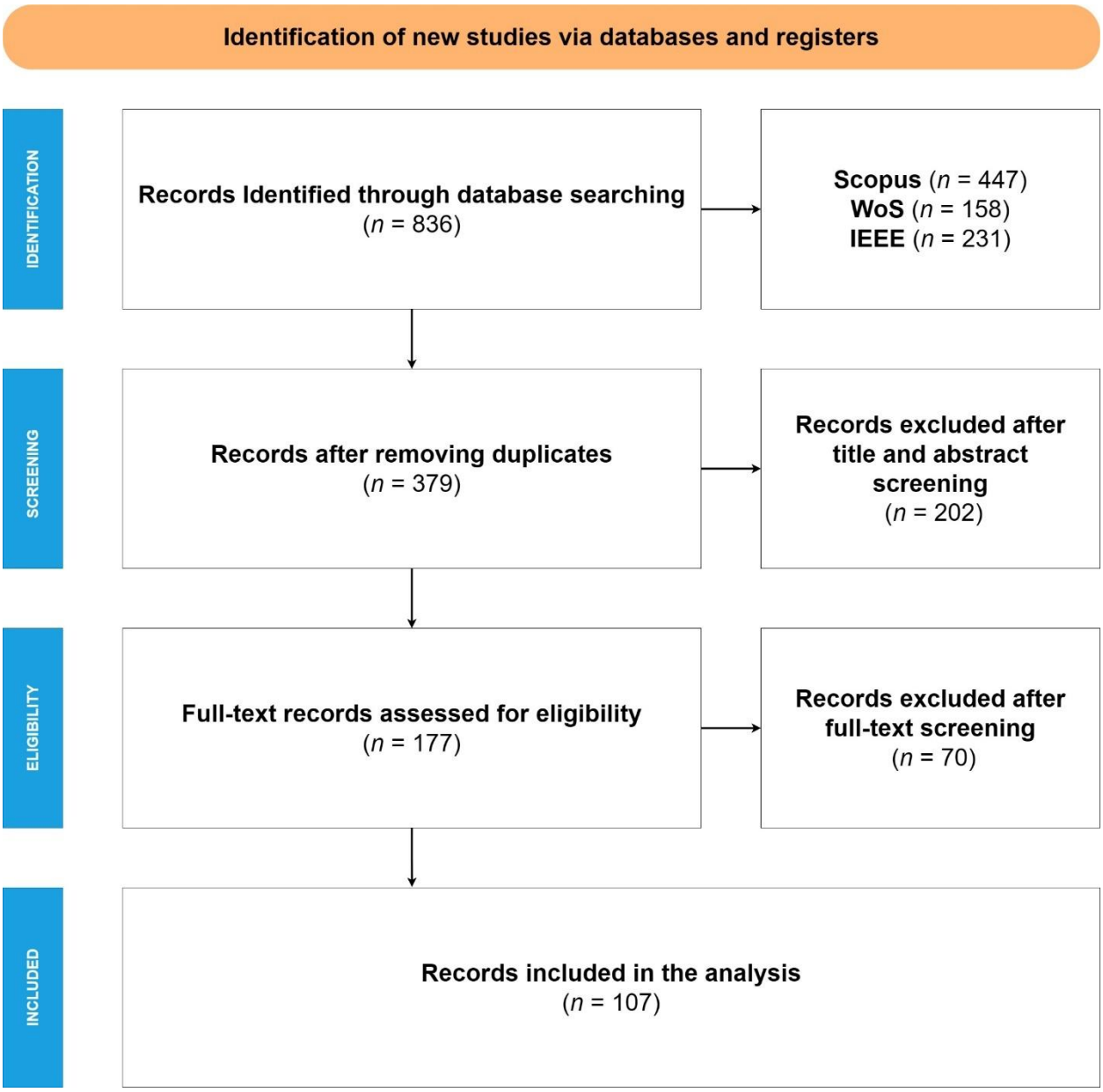

Figure 1. PRISMA Flow Diagram for the Study Selection

RESULTS

The search across databases focusing on the application of ChatGPT in programming education yielded 836 records. This number is understandably smaller compared to broader literature reviews. For comparison, a systematic review uncovered 1,091 studies that examined the integration of ChatGPT into nursing education contexts (e.g., Kucukkaya et al., 2024). The larger number of studies in the prior review remains higher compared to the present study, which highlights the more specialized focus on programming education. In adherence to PRISMA

MANUEL B. GARCIA
www.manuelgarcia.info

guidelines (see Figure 1), the present study selected 107 relevant documents. Within this refined dataset, most documents were published as conference papers (n = 73, or 68.22%), with journal articles comprising a smaller portion (n = 34; 31.78%). These documents were disseminated across 81 distinct sources and authored by 394 contributors, with an average collaboration rate of 4.62 authors per document. Collectively, these papers have accumulated 724 citations, yielding an average citation rate of 6.77 per document. Given that ChatGPT was introduced in 2022, it is anticipated that the surge in publications primarily occurred in 2023 and beyond.

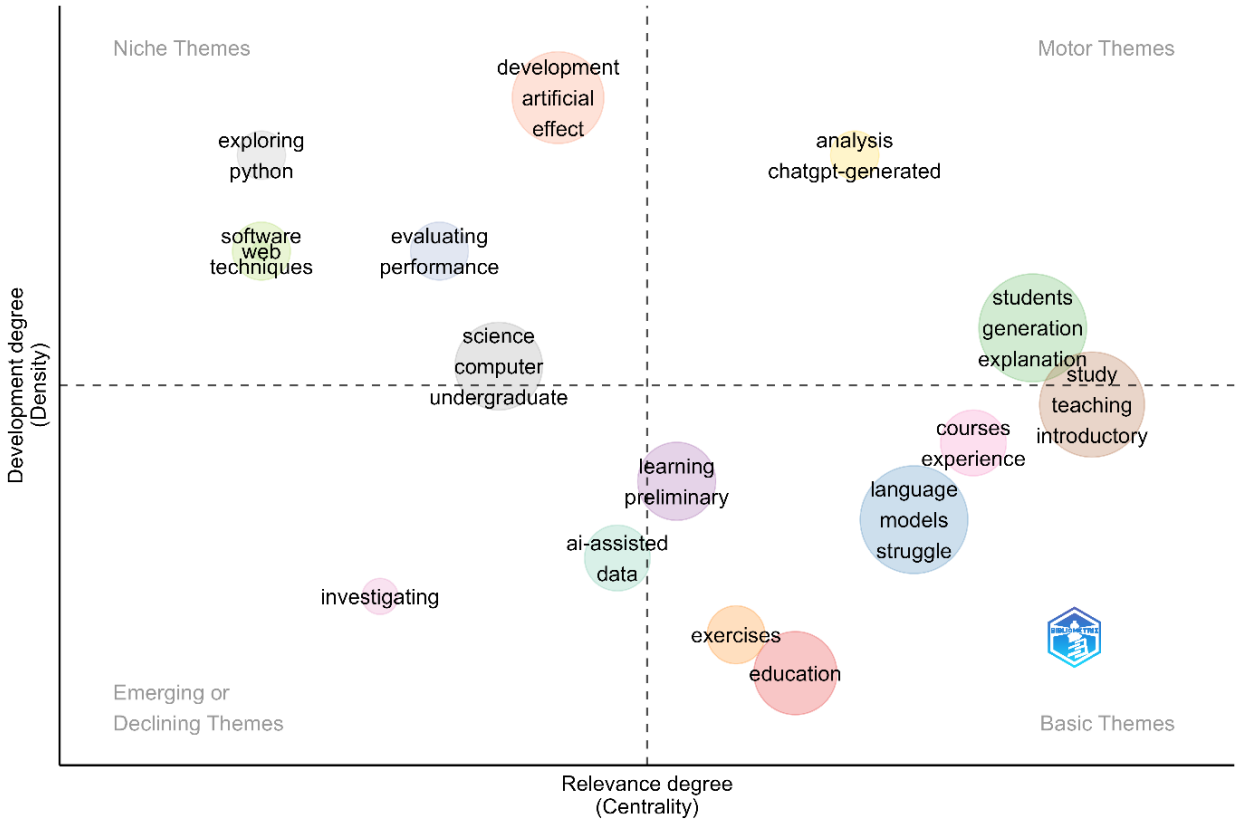

Figure 2. Strategic Diagram of the Composite Thematic Map

RQ1: What are the trending research topics related to ChatGPT in programming?

When analyzing bibliographic data, Garcia (2023b) asserted that the identification of themes and topics is the most critical insight of conceptual analysis through co-word occurrence. In a linguistic sense, co-word occurrence refers to analyzing how often specific terms appear together in texts, which helps identify relationships between different research topics and their relative importance within a field. This method is particularly insightful as it illuminates which aspects of research on ChatGPT in programming are immature, established, declining, and emerging. Consequently, future research can pinpoint which areas of study need further exploration. Depicted via a composite thematic map (CTM), Figure 2 highlights the conceptual framework of the dataset. CTM is a visual tool used in bibliometric analysis to categorize research

topics based on how central and well-developed they are within the field. It organizes research themes into four quadrants based on their centrality (i.e., degree of interaction) and density (i.e., internal strength). Keywords with the highest frequency of occurrence are represented as bubbles, placed according to the centrality and density metrics of their corresponding themes. In the top right quadrant, *'analysis'* and *'chatgpt-generated'* are identified as Motor Themes. These themes signify well-established and influential areas within the research field. The top left quadrant highlights Niche Themes, such as *'development,' 'artificial,'* and *'effect,'* alongside specific terms related to *'python'* and *'software.'* These themes are detailed and mature in their development yet remain less connected to the central discourse of the field. Basic Themes such as *'students,' 'teaching,'* and *'education'* are found in the bottom right quadrant. These themes represent fundamental and evolving areas that are starting to play a more central role in research. Lastly, the bottom left quadrant reveals Emerging or Declining Themes such as *'data,' 'ai-assisted,'* and *'investigating.'* These themes indicate research areas that are either in the nascent stages of development or are diminishing in focus within the broader research context.

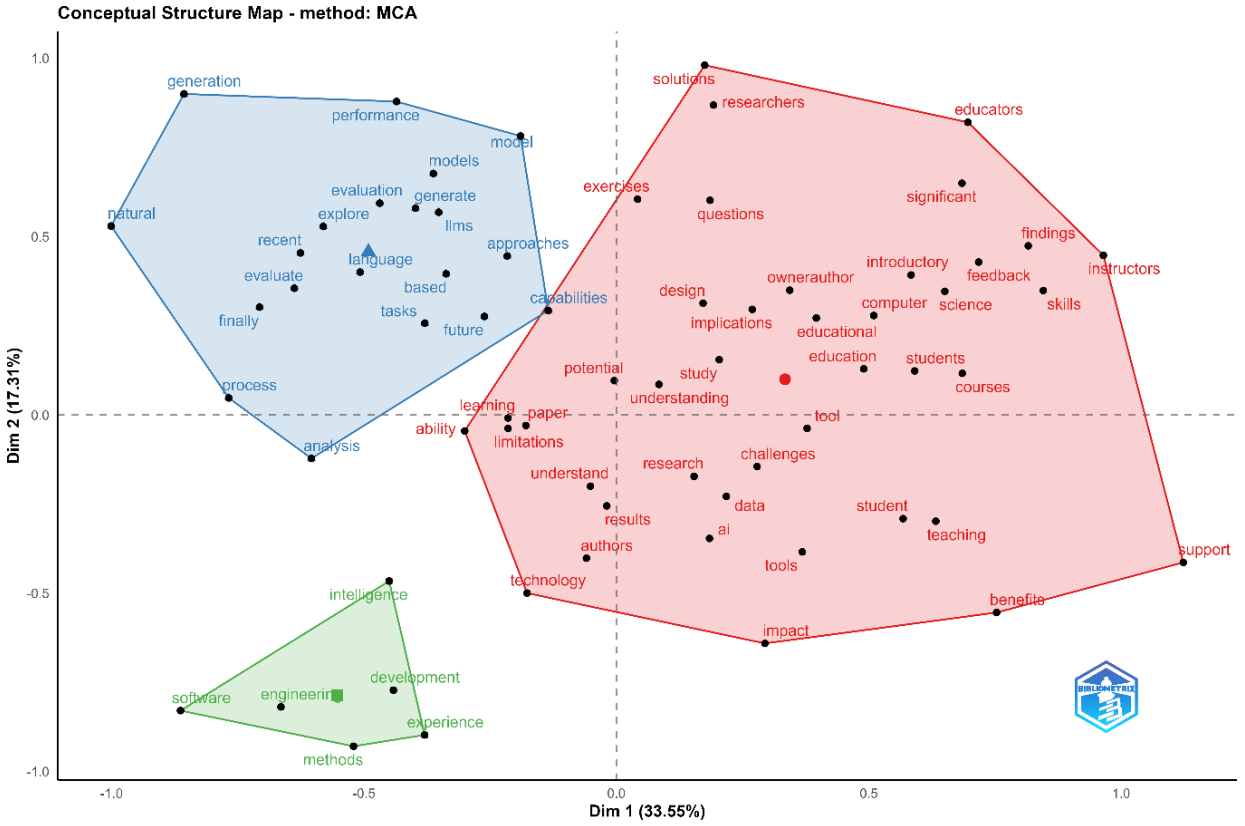

Figure 3. Conceptual Structure Map Using Multiple Correspondence Analysis

In conjunction with CTM, the present study likewise constructed a conceptual structure map (CSM) employing Multiple Correspondence Analysis (MCA). The CSM serves as another visualization tool that elucidates the interconnections among various concepts within the research. In contrast to the CTM, which categorizes themes based on their internal density and external centrality, the CSM identifies broader clusters within the research focus areas. The development

of a CSM often involves the application of MCA, as suggested by prior studies (e.g., Garcia, 2023b). MCA is a technique for dissecting multivariate categorical data to unearth underlying patterns and relationships. This method, when used in tandem with k-means clustering, facilitates the formation of concept clusters. As depicted in Figure 3, the resultant CSM for this dataset has automatically categorized the intellectual landscape and research focal points of ChatGPT within programming education into three primary clusters. The largest cluster is highlighted in blue, which encompasses a range of terms that primarily relate to the educational domain. Keywords such as *'students,' 'courses,' 'education,'* and *'teaching'* signal a strong research emphasis on pedagogical aspects (e.g., Zheng, 2023). This cluster signifies the broad impact and considerations of ChatGPT in educational settings. The second cluster is highlighted in blue, which appears to concentrate on evaluative aspects of research, involving terms like *'evaluation,' 'performance,' 'models,'* and *'tasks.'* This cluster suggests a focus on assessing the effectiveness, outcomes, and applications of ChatGPT in programming education, examining both the process and the tools involved (e.g., Piccolo et al., 2023). The third cluster, marked in green, is associated with keywords such as *'software,' 'engineering,'* and *'development.'* This indicates a cluster that is centered around the developmental aspects of ChatGPT, possibly exploring the technological underpinnings, software engineering perspectives, and the progression of ChatGPT's capabilities in educational programming environments (e.g., Kuramitsu et al., 2023).

RQ2: How does ChatGPT assist in teaching and learning programming?

Personalized Tutoring

Numerous research efforts have pinpointed various instructional strategies where ChatGPT serves to bolster computer programming education. A notably prevalent strategy is its use in personalized tutoring (Wieser et al., 2023), which is in line with the concept of a "*More Knowledgeable Other*" (MKO). Within Vygotsky's notion of the Zone of Proximal Development (ZPD), an MKO refers to someone possessing more knowledge or skill in a given domain. Importantly, this role is not limited to humans (Jarrett, 2022), allowing ChatGPT to serve as an MKO. As a virtual tutor embodying this concept, ChatGPT offers individualized feedback and detailed explanations (Randall et al., 2024). It can scrutinize student-written code, identify mistakes, and provide context-specific guidance to enhance their coding proficiency (Chen et al., 2023; Kosar et al., 2024). This personalized interaction not only facilitates immediate error correction but also encourages students to refine their coding techniques (Leinonen et al., 2023). Moreover, it helps in understanding best practices and developing a more robust grasp of programming concepts. This approach fosters a tailored, supportive learning experience that leverages the principles of scaffolded learning and self-directed programming (Sun et al., 2024). The assertion of Garcia (2023a) on the critical importance of instructional guidance, notably absent in many computer science courses, underscores the value of integrating such AI-driven personalized tutoring. By assuming the role of an additional MKO, ChatGPT can provide individualized support to students, a task that would otherwise be logistically challenging for teachers to offer in classrooms with high student-to-teacher ratios. In a more innovative example, Yang et al. (2024) utilized *PyTutor*, a ChatGPT-based Intelligent Tutoring System, and observed that students with lower initial knowledge exhibited significantly higher engagement, improved completion rates, and greater success rates in both in-class and after-class programming exercises.

This approach underscores the potential of integrating ChatGPT with adaptive technologies to address diverse learner needs and enhance overall learning outcomes.

Knowledge Reinforcement

While some research suggests that AI code generators may not scale efficiently as a teaching and learning tool (Popovici, 2023), other studies have noted its effectiveness in the programming instruction process (Husain, 2024; Kazemitabaar et al., 2023). For instance, a common learning difficulty in programming is understanding abstract concepts, which can be challenging for novice programmers and students to grasp only through theoretical explanations (Garcia, 2024; Tsai, 2019). Although teachers address these topics in the classroom, students can engage with ChatGPT outside class hours. This practice serves as a reinforcement to what they have learned and provides an additional opportunity to clarify any persisting doubts or to solidify their understanding of complex topics. Supporting this approach, Yilmaz and Yilmaz (2023) and Sun et al. (2024) noted that ChatGPT can provide programming examples with clear explanations as well as resources on advanced topics that can significantly contribute to the learning reinforcement process. It can also be used to aid memory recall, as the active retrieval practice helps to strengthen memory retention (Bai et al., 2023). By reinforcing learning outside of the classroom, teachers can ensure that students have a more robust understanding of the material, which can lead to more productive classroom sessions. When students come to class with a clearer grasp of programming concepts, teachers can spend less time re-explaining foundational topics and more time on advanced material and interactive learning experiences.

Instructional Materials

Another advantage of using ChatGPT in teaching and learning programming is its capability to generate customized instructional materials (Bringula, 2024; Hartley et al., 2024; Husain, 2024). Teachers can effortlessly generate a wide array of lesson plans, coding exercises, programming quizzes, and example projects tailored to their curriculum's specific needs and difficulty levels (Zheng, 2023). It also allows them to diversify their teaching materials and provide students with a broad spectrum of coding challenges that cater to various skill levels and learning styles. Further supporting this perspective, Speth et al. (2023) conducted a study on the deployment of ChatGPT-generated exercises within beginner and intermediate Java programming courses. Their findings indicate that the exercises produced by ChatGPT are not only suitable for academic settings but also indistinguishable from those created by humans. This finding aligns with research conducted by Spasić and Janković (2023), who assessed the capability of ChatGPT in devising programming lesson plans for preschoolers. They observed that its ability to mimic human-like responses is remarkably extensive. In each of the three prompt scenarios they examined, the responses were not only relevant to the assigned task and theme but also resulted in lesson plans that adhered closely to established literature guidelines. The structure of these lesson plans likewise conformed to recommended best practices, encompassing clear objectives, necessary materials, and detailed lesson outlines. Gutierrez et al. (2024) reported similar findings after testing ChatGPT's ability to generate programming exercises. Their results consistently featured high-quality machine problem descriptions, accurate code solutions, and well-structured

code. This finding suggests that students can be exposed to a wider range of problems, providing them with additional practice to improve their programming knowledge and skills.

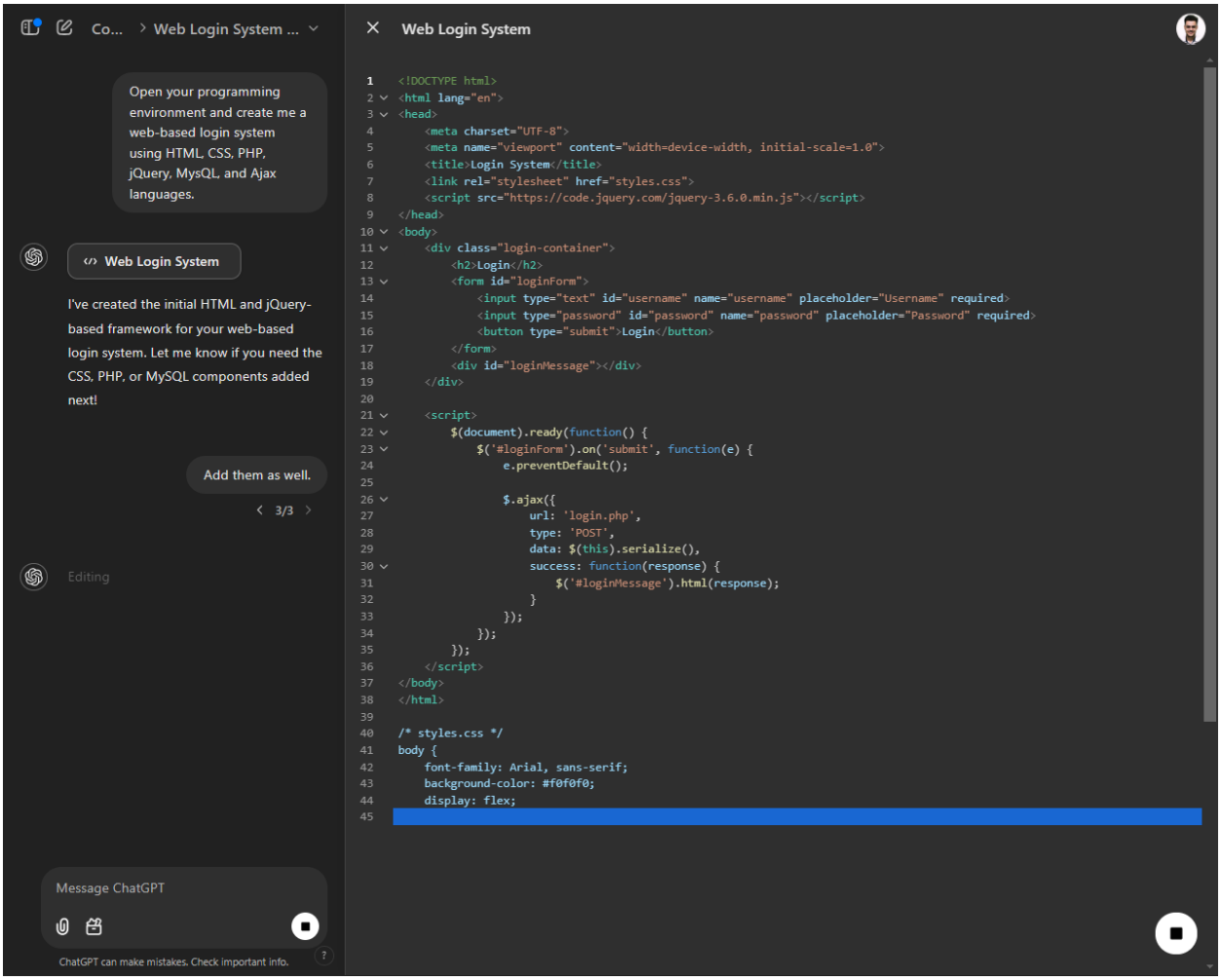

Figure 4. ChatGPT Generating Source Code using the New Canvas Interface

Source Code Generation

The capability of ChatGPT to generate code is regarded as a transformative tool for programmers, educators, and students. Recently, the release of the *canvas* feature has made coding in ChatGPT even more accessible and intuitive (see Figure 4). This new addition allows users to seamlessly edit and iterate on code in a collaborative environment. However, the automatic code generation feature is often met with criticism. For instance, Bringula (2024) argues that this capability may encourage overdependence on this technology and potentially undermine the development of critical thinking and problem-solving skills. There are also concerns regarding academic dishonesty (Sun et al., 2024), as students may be tempted to submit AI-generated code as their own (Rose et al., 2023). Despite these concerns, ChatGPT can provide substantial educational benefits when used with integrity. For example, Jacques (2023) posited

that students can use ChatGPT to produce a variety of solutions and compared the differences in efficiency, approach, and style. In educational research, there is substantial evidence supporting the benefits of learning from and comparing across multiple examples (Fiorella, 2023). This comparison can also extend to juxtaposing AI-generated code with that crafted by humans (Garcia et al., 2023), which would offer insights into the readability and maintainability of code. Moreover, students can be encouraged to use AI to generate an initial solution and then challenge themselves to devise an alternative code independently. As an example, after ChatGPT provides a Python script to sort a list of numbers, students could create a different sorting algorithm manually. Another use of AI-generated code involves educators prompting students to describe what it does and why it works. Corney et al. (2014) and Vieira et al. (2017) suggested that by requiring students to articulate the functionality and rationale behind the code, they are compelled to engage in critical thinking about the code's purpose and its operational mechanisms. In the context of big data analytics education, Park and Kim (2025) added that students utilizing ChatGPT outperformed their peers who relied on Stack Overflow or worked without any external support.

Immediate Feedback

Immediate feedback plays a pivotal role in the scaffolding process for effective learning, particularly in the context of programming education (Garcia, 2021; Shaka et al., 2023). The promptness with which learners receive feedback on their programming efforts is crucial for their development. Marwan et al. (2020) highlighted the significance of this immediacy, noting that quick feedback can significantly enhance the engagement of novice programmers and bolster their determination to continue their studies in computer science. Expanding on this concept, Yilmaz and Yilmaz (2023) pointed out the advantages of using ChatGPT in providing instantaneous feedback, which can accelerate the learning process and enhance student comprehension. At the very least, the ease of use of ChatGPT can encourage engagement from students who might typically be reluctant to seek help (Maher et al., 2023). Such an enhancement in programming education is attributed to the ability of ChatGPT to offer clear explanations of code and support in debugging (Chen et al., 2023; Randall et al., 2024). Furthermore, Sun et al. (2024) highlighted that students can directly input their code snippets or detailed error messages into ChatGPT, which significantly streamlines the acquisition of tailored feedback. Their research found that the provision of such personalized feedback has been instrumental in enhancing students' programming education. Such immediate and customized feedback additionally aids teachers by enhancing the efficiency of the feedback process (Rose et al., 2023). This optimization allows educators to ensure that students receive precise guidance tailored to their specific coding challenges without the need for extensive manual review. By integrating ChatGPT into the feedback loop, teachers can allocate more time to other critical aspects of education.

Assessment Assistance

Evaluating student code is a complex and time-consuming task. Teachers often face significant challenges in this area, as they must not only determine the correctness of the code but also assess its efficiency, readability, and adherence to best practices (Garcia et al., 2022). ChatGPT transcends its role in generating assessments by offering substantial aid in the evaluation process itself. In corroborating this idea, Wieser et al. (2023) investigated the

effectiveness of ChatGPT's method for assessments. Their findings indicated that the distribution of grades by ChatGPT was uniform and logical, accompanied by further explanations that shed light on the foundational aspects of its grading criteria. Moreover, its capability to scrutinize code enables a detailed analysis that encompasses not just the detection of errors but also assessments of the code's optimization, clarity, and alignment with professional coding standards (Chen et al., 2023; Yan et al., 2023). Jukiewicz (2024) added that, unlike teachers who require several hours to grade assessments, ChatGPT can complete the grading process much faster, approximately in 9.5 seconds for a single query. There is also a positive aspect for students in using ChatGPT to assess their own work. Li et al. (2023) observed that the code produced by ChatGPT tends to align more closely with established coding conventions than that written by humans. For example, when coding in C++ programming language, ChatGPT typically opts for the standard "*endl*" line terminator instead of the "*\n*" newline character. It also prioritizes readability by using descriptive words for identifiers, as opposed to humans who might opt for abbreviations or letters that can be unclear in their meaning. When students receive such constructive feedback from ChatGPT and follow these exemplars of clarity and convention in their coding activities, it can lead to submissions that are easier for teachers to assess (Husain, 2024).

RQ3: What issues arise from using ChatGPT in programming instruction?

Academic Dishonesty

Despite the numerous potential benefits of integrating ChatGPT into programming instruction, various studies have identified challenges and issues that can emerge. Academic dishonesty, particularly in the form of code plagiarism, stands out as a prevalent issue that could undermine the integrity of programming education (Akçapınar & Sidan, 2024; Lau & Guo, 2023; Rose et al., 2023). As noted by Chen et al. (2024), the use of generative AI tools in coding has contributed to a noticeable rise in plagiarism cases. More specifically, there has been a shift from '*suspected online plagiarism hubs*' like Chegg or CourseHero to ChatGPT. This concern stems from the ease with which students can generate source code using ChatGPT, potentially leading to instances where the work submitted is not genuinely their own. Such a situation complicates the assessment of students' actual understanding and capability in coding, as it blurs the lines between learning assistance and outright plagiarism (Budhiraja et al., 2024; Gasiba et al., 2023). Even seasoned instructors may struggle to detect AI-generated code, making it harder to assess students' true abilities (Ellis et al., 2024). The availability of instant solutions can also tempt students to bypass the critical thinking and problem-solving processes essential for their development as competent programmers. As students become used to code generation, Popovici (2023) argued that it is imperative to develop new programming competencies. For instance, teachers may educate students on how to review AI-generated code for errors, optimize it for efficiency, and ensure it meets project requirements and coding standards. Such competencies ensure that students can effectively use AI as a tool rather than a substitute. Savelka et al. (2023) also asserted the importance of cultivating an educational environment where the emphasis is placed on learning and personal development rather than solely on achieving the correct answers. Emphasizing the importance of academic integrity and ethical conduct in the classroom is especially crucial in an era of LLMs and generative AI applications (Bozkurt et al., 2024).

Ethical Issues

Ethical concerns represent a consistent theme across various research studies (Liu, 2023; Petrovska et al., 2024; York, 2023). This focus on ethics underscores the importance of addressing moral questions that arise in the context of using technologies like ChatGPT in education. These issues encompass a range of considerations, from the origins and reliability of the generated code to the imperative of nurturing learner autonomy to avoid harmful reliance. Silva et al. (2024) emphasized the significance of promoting a profound and self-sufficient grasp of programming principles, reflecting the broader ethical challenge of ensuring students develop their skills without undue dependence on AI tools. According to Feng et al. (2023), there have been specific instances where ChatGPT was tasked with creating a Python function to predict an individual's seniority or evaluate their scientific competence. Unfortunately, these instances revealed that the code produced by ChatGPT could manifest biases related to demographic factors, raising serious ethical questions about the objectivity and fairness of AI-generated content. This issue aligns with broader concerns about AI models inheriting biases present in their training data, which can unintentionally perpetuate stereotypes or discriminatory outcomes. Such biases challenge the ethical deployment of AI in sensitive educational contexts, where fairness and equity are paramount. Additionally, concerns surrounding data privacy and the potential for legal repercussions further complicate the ethical landscape. Valový and Buchalcevova (2023) pointed out that the integration of ChatGPT into educational practices introduces risks related to the handling and protection of sensitive information, underscoring the necessity for robust data governance measures. All these studies posited that addressing these challenges requires a concerted effort to implement ethical guidelines and practices that respect the integrity of the learning process while harnessing the potential of AI tools like ChatGPT.

Overreliance on ChatGPT

There is empirical evidence that ChatGPT is a proficient coder (Chen et al., 2023; Li et al., 2023; Ouh et al., 2023; Piccolo et al., 2023). This code generation capability, however, risks becoming a double-edged sword as students might increasingly turn to ChatGPT for swift solutions (Budhiraja et al., 2024; Jošt et al., 2024). Rather than engaging with programming problems independently or leveraging critical thinking skills acquired in the classroom, there is a tendency among students to depend on ChatGPT for immediate answers (Joshi et al., 2024). Teachers have expressed concerns that students may gain fewer programming skills and experience a decline in both learning and code quality due to their overreliance on ChatGPT (Groothuijsen et al., 2024). This pattern of dependency could escalate to a stage where students consult ChatGPT for solutions to tasks they are well-equipped to solve by themselves. As pointed out by Silva et al. (2024), the issue of becoming overly dependent on generative AI tools for code generation is concerning. The convenience provided by ChatGPT might cause students to neglect nurturing essential programming skills. Kazemitabaar et al. (2023) observed that excessive reliance on such tools could hinder the learning process. As a potential countermeasure, they suggested implementing measures that require learners to interact with the AI-generated code in educational ways before using it directly. For example, incorporating a preliminary task such as tackling a Parsons problem based on the AI-generated code or responding to multiple-choice queries that cover the concepts present in the generated code (e.g., grasping the concept of nested loops),

could promote an active learning environment. These initial activities aim to immerse students in the subject matter prior to their use of AI-generated solutions, presenting creative methods to boost active learning and improve students' understanding in a customized fashion.

Impaired Critical Thinking

A detrimental consequence of excessive dependence on ChatGPT is the potential degradation of critical thinking skills (Bai et al., 2023). Critical thinking is vital in programming, especially as it equips students with the ability to methodically approach problem-solving (Garcia, 2023a). With ChatGPT providing instantaneous solutions, it poses a risk of students relying on these answers without fully understanding the foundational principles they are based on. Such reliance can lead to a superficial understanding of key programming concepts, as genuine critical thinking involves not only identifying solutions but also grasping the reasons behind them. Furthermore, becoming a proficient programmer is inherently linked to the experience of making mistakes and mastering the art of debugging. Relying on ChatGPT to consistently provide corrections or suggest error-free code may hinder the development of essential debugging skills in students, which demand an in-depth comprehension of the code, logical reasoning, and attention to detail. Husain (2024) pointed out that instructors need to put in extra effort to design programming assignments that necessitate the application of programming knowledge and critical thinking rather than setting straightforward or trivial tasks that could be easily completed through direct queries. Sharing the same perspective, Ellis et al. (2024) recommended making assessment prompts more general. This approach aims to challenge students who lack a programming understanding, preventing them from achieving a passing score by simply inputting the assignment instructions into ChatGPT. The goal is to encourage students to engage critically with their programming tasks. Additionally, Yilmaz and Yilmaz (2023) recommended assigning complex and modular programming challenges to encourage students to synthesize complete solutions from modular responses. They argue that tackling complex and unstructured problems plays a pivotal role in cultivating students' computational thinking abilities (Cheng et al., 2023).

Technical Limitations

Drawing parallels with findings from various disciplines (Dave et al., 2023) and reviews (Cong-Lem et al., 2024), it is evident that ChatGPT shares similar technical limitations when applied to programming instruction. For instance, its text-based nature and lack of specialized programming features, unlike those found in Integrated Development Environments (IDEs), present significant challenges. The absence of these coding features in ChatGPT means it cannot offer the same level of support for the iterative and experimental processes that are fundamental to programming. Researchers such as Ouh et al. (2023), Yilmaz and Yilmaz (2023), Shaka et al. (2023), and Budhiraja et al. (2024) have further highlighted the issue of accuracy in the code generated by ChatGPT. They pointed out that while ChatGPT can produce code quickly, it sometimes lacks the precision and reliability required for programming tasks. This inconsistency can mislead learners, especially novices, into adopting incorrect programming practices or misunderstanding fundamental concepts. Expanding on this limitation, Piccolo et al. (2023) emphasized ChatGPT's shortcoming as a non-IDE tool, specifically its inability to execute code. Therefore, ChatGPT often cannot predict code outcomes, underlining the indispensable role of

human feedback in the learning process. This gap highlights a critical area where direct interaction with code and immediate output observation, features intrinsic to IDEs, are vital for comprehensive programming education. Lastly, DePalma et al. (2024) highlighted that ChatGPT faces challenges in understanding the broader context in which individual code segments are applied. These scenarios often lead ChatGPT to offer suggestions based on misinterpretations or incorrect assumptions about the overall structure or functionality of the code. Consequently, it may sometimes propose solutions that are irrelevant or unnecessary.

DISCUSSION AND IMPLICATIONS

Propelled by advancements in AI technologies, technology-enhanced programming education has witnessed remarkable growth recently. The introduction of ChatGPT has opened new avenues for exploration, with numerous studies beginning to investigate its potential implications and applications (Bringula, 2024; Silva et al., 2024). Despite the growing interest, there is a notable research gap in understanding the scope of ChatGPT's integration, utilization, and impact within programming education. This rapid review seeks to bridge this gap by examining the current literature to identify trending research topics surrounding the use of ChatGPT in computer programming, evaluate how ChatGPT supports teaching and learning in programming, and discern the challenges that emerge from its application in programming instruction. As highlighted in the reviewed studies, teachers and students are already leveraging ChatGPT for their programming activities, making it impractical to prohibit its use entirely. Despite calls for limiting or outright banning its access (Lau & Guo, 2023), the trend unmistakably suggests that ChatGPT will remain a fixture in the educational landscape.

To illustrate the current literature surrounding the use of ChatGPT in programming, trending research topics were identified, highlighting key areas of focus and development. Based on the results, it is noticeable that the identified topics and themes reflect a broader historical evolution of AI integration into education. First, the motor themes emerging from the literature indicate the growing significance of AI-generated content and automated analysis tools (Chen et al., 2023; Randall et al., 2024), which have become central to modern educational practices. Historically, AI began as a niche area with limited application in education, primarily focused on adaptive learning systems and intelligent tutoring. However, as AI technologies matured, particularly with the advent of LLMs like ChatGPT, their role expanded, influencing curriculum development (Speth et al., 2023), student assessments (Wieser et al., 2023), and personalized learning experiences (Randall et al., 2024). Meanwhile, the Niche Themes reveal that while AI techniques are becoming increasingly specialized and sophisticated, their integration into broader educational discourse remains limited. The current literature suggests that certain advanced applications of AI, particularly in technical areas such as programming and software development, are highly developed but remain peripheral to the broader educational landscape. For example, while ChatGPT is being used by some researchers and teachers in experimental or supplemental capacities, as well as by students outside formal classroom settings, it has yet to be fully embedded into general education settings (e.g., introductory computer science courses; Mahon et al., 2024). This current setup indicates that although the technology is being explored, its systematic integration into formal education frameworks remains limited.

In terms of teaching and learning, the implications of integrating ChatGPT into programming education are multifaceted. From an educational curriculum standpoint, the integration of ChatGPT into programming education necessitates a thoughtful revision of existing curricula (Randall et al., 2024). This necessity for curricular adaptation stems from the evolving nature of programming and computer science fields, where AI and machine learning are becoming increasingly central. Integrating ChatGPT and other emerging generative AI tools and LLMs into the curriculum can provide students with a more contemporary learning experience that mirrors the real-world applications and challenges they are likely to encounter in their careers. Moreover, as the job market for future programmers and developers is expected to demand proficiency in AI competencies and prompt engineering skills, curriculum designers are urged to weave AI literacy, ethics, and a comprehensive understanding of AI's societal, legal, and security implications into programming courses. However, it is important to note that while these tools may enhance the learning experience, they do not necessarily guarantee improved academic performance, as studies have shown differing results regarding their effectiveness in fostering deeper understanding and skill mastery. For example, Yilmaz and Yilmaz (2023) reported that using ChatGPT in programming instruction can improve students' coding skills, whereas Sun et al. (2024) found no statistically significant difference between students utilizing ChatGPT and those engaged in self-directed, facilitated programming. Empirical evidence also shows that ChatGPT performs well with easy and medium coding problems but struggles with more complex tasks (Bucaioni et al., 2024). Nevertheless, this broadened educational approach is vital in equipping students with the diverse skill set needed to navigate the rapidly changing landscape, marked by the emergence of new AI tools and their deep integration into various business processes.

For teachers, the practical implications involve designing educational strategies that harness the potential of ChatGPT to enhance standard teaching practices without compromising the foundational principles of self-guided learning and critical reasoning. While ChatGPT offers advantages such as tailored instruction and automated code creation, it also poses the risk of fostering dependency and diminishing critical thinking skills. Moreover, some evidence suggests that students show a preference for teacher-created explanations over AI-generated ones in fundamental programming concepts, such as sequence, selection, and iteration (Lee & Song, 2024). Consequently, teachers are advised to judiciously integrate AI-based educational tools and resources into their teaching frameworks. For example, teachers can improve students' comprehension of course material by presenting programming demonstrations that feature ChatGPT. It is also crucial for teachers to vigilantly check for signs of academic misconduct. Teachers may use software like the *Measure of Software Similarity* (MOSS) and *Codequiry* to aid in identifying potential code plagiarism (Qureshi, 2023). In the absence of robust plagiarism tools, alternative methods can be implemented. Popovici (2023) suggested requiring students to modify small, singular lines of code under teacher supervision to earn points. In this approach, teachers may deliberately introduce a compile error in one line of code and challenge students to rectify it. Another effective strategy involves requiring students to annotate their code with comments (Ellis et al., 2024). Mandating specific commentary throughout their programming assignments can assist teachers in discerning students' understanding and originality, thereby enhancing the educational value of assignments and promoting academic integrity.

For students, the integration of ChatGPT into programming education opens the door to an expansive reservoir of knowledge and assistance. This tool can instantly clarify doubts, provide explanations for complex programming concepts, and even generate code examples. However, this advantage comes with the essential condition of upholding academic integrity and a deep-seated commitment to genuinely grasp the fundamental principles of programming. The implications of this scenario for students are multifaceted. On the one hand, ChatGPT serves as a powerful educational ally that potentially democratizes access to high-quality programming education by making it more accessible and personalized. Students who might otherwise struggle with traditional learning resources or face barriers due to geographical and economic constraints can benefit from immediate, tailored support. On the other hand, the ease of access to solutions poses a risk of diminishing the incentive for students to engage deeply with the material. The temptation to use ChatGPT as a shortcut, bypassing the effortful process of coding from scratch, debugging, and internalizing the logic behind programming tasks, could impair the development of critical thinking and problem-solving skills. These skills are not only vital for academic success but are also indispensable in the real-world programming environment, where understanding the 'why' behind the 'how' is crucial (Berrezueta-Guzman & Krusche, 2023). Furthermore, the reliance on ChatGPT for immediate answers could potentially undermine the development of perseverance and resilience—qualities that are nurtured through facing and overcoming challenges. Programming is as much about dealing with frustration and learning from failure as it is about celebrating success. Encountering and working through errors is a fundamental part of the learning process that fosters growth, adaptability, and innovation.

Overall, while ChatGPT offers substantial benefits in programming education, the limitations also necessitate a balanced approach to its use. The education sector must navigate this new terrain carefully to ensure that the technology serves as a complement to traditional learning methods rather than a replacement (Bozkurt et al., 2024). Emphasizing the importance of academic integrity, fostering a culture of honest inquiry, and encouraging students to explore beyond surface-level understanding are crucial steps in leveraging ChatGPT's potential while safeguarding the integrity and depth of programming education. Given that this study constitutes a rapid review, it is imperative that future research endeavors undertake a more comprehensive systematic review and meta-analysis, utilizing a more developed corpus of literature. As the academic discourse around ChatGPT and its implications for programming education continues to evolve, a systematic review will offer a more detailed understanding of its impact. Such research efforts should ideally be pursued once the literature on ChatGPT in programming education becomes more mature. This future research will be critical in guiding educators, policymakers, and curriculum developers in making informed decisions about the integration of AI tools like ChatGPT into educational frameworks. Nevertheless, it remains uncertain whether scholarly interest in using ChatGPT for programming instruction is expected to maintain an upward trajectory in subsequent years. This uncertainty only strengthens the necessity for future research to re-examine the literature to ascertain the trajectory of publication trends. The academic community must maintain a continuous progression in research to enhance our comprehension of this tool and to effectively tackle emerging challenges. By consistently updating the educational field with fresh empirical evidence, the benefits of ChatGPT can be more efficiently exploited.

Another research avenue identified in this review pertains to the need for more specific, targeted studies that address the current gaps in ChatGPT's use within programming education. Several studies have highlighted vital research areas that require further exploration, such as the impact on teaching effectiveness (Bringula, 2024), the influence on students' engagement and class collaboration (Husain, 2024), methods for enhancing prompt utilization (Sun et al., 2024), and the development of ethical guidelines for its use (Petrovska et al., 2024). To advance the field more directly, future research could focus on experimental or observational studies designed to answer specific questions raised by these themes. For example, an experimental study could examine how different instructional designs that integrate ChatGPT affect teaching effectiveness, comparing various pedagogical approaches and assessing their impact on learning outcomes. Additionally, observational studies could explore how ChatGPT influences student engagement and collaboration within group-based programming tasks, examining both in-class and out-of-class interactions. Future research could also involve design-based research aimed at refining prompt engineering techniques, where iterative cycles of testing and refinement are used to optimize how educators and students interact with ChatGPT to enhance learning experiences. There is also a growing need for qualitative research to explore the ethical implications of ChatGPT's deployment in education. This could involve interviews or focus groups with educators, students, and policymakers to develop ethical frameworks and guidelines that address concerns such as academic integrity, data privacy, and bias in AI-generated content. By addressing these specific research questions and employing diverse methodologies, future studies can provide deeper insights into ChatGPT's role in programming education and contribute to the development of effective, ethical, and impactful AI-enhanced teaching and learning practices.

CONCLUSION

This rapid review into the utilization of ChatGPT in programming education reveals a research landscape teeming with potential yet fraught with issues. By scrutinizing the current literature, this study has illuminated the promising role of ChatGPT in enhancing programming instruction. Its applications include providing personalized tutoring, reinforcing knowledge, creating instructional materials, generating source code, providing immediate feedback, and aiding in assessment. These uses underscore ChatGPT's capacity to revolutionize programming education by enhancing both teaching and learning experiences. However, the adoption of this technology is not without concerns, such as the potential for academic dishonesty, ethical dilemmas, the danger of reducing critical thinking skills, becoming too dependent on technology, and facing technical hurdles. The insights derived from this review necessitate a measured and thoughtful integration of ChatGPT within programming curricula. Furthermore, the recommendations and implications laid out herein serve as a roadmap for stakeholders across the educational spectrum. As we continuously explore the integration of ChatGPT into programming education, it is imperative that further evaluation, adaptation, and dialogue among all educational stakeholders take place. This collaborative effort will pave the way for a balanced and forward-thinking approach to incorporating AI into our educational frameworks.

REFERENCES

Akçapınar, G., & Sidan, E. (2024). AI Chatbots in Programming Education: Guiding Success or Encouraging Plagiarism. *Discover Artificial Intelligence*, *4*(1), 1-10. https://doi.org/10.1007/s44163-024-00203-7

Aria, M., & Cuccurullo, C. (2017). bibliometrix: An R-tool for Comprehensive Science Mapping Analysis. *Journal of Informetrics*, *11*(4), 959-975. https://doi.org/10.1016/j.joi.2017.08.007

Asian Development Bank. (2022). Digital Jobs and Digital Skills: A Shifting Landscape in Asia and the Pacific. https://doi.org/10.22617/SPR220348

Bai, L., Liu, X., & Su, J. (2023). ChatGPT: The Cognitive Effects on Learning and Memory. *Brain-X*, *1*(3), 1-9. https://doi.org/10.1002/brx2.30

Baig, M. I., & Yadegaridehkordi, E. (2024). ChatGPT in the Higher Education: A Systematic Literature Review and Research Challenges. *International Journal of Educational Research*, *127*, 1-20. https://doi.org/10.1016/j.ijer.2024.102411

Berrezueta-Guzman, J., & Krusche, S. (2023). Recommendations to Create Programming Exercises to Overcome ChatGPT. *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T)*, 147-151. https://doi.org/10.1109/CSEET58097.2023.00031

Bozkurt, A., Xiao, J., Farrow, R., Bai, J. Y. H., Nerantzi, C., Moore, S., Dron, J., Stracke, C. M., Singh, L., Crompton, H., Koutropoulos, A., Terentev, E., Pazurek, A., Nichols, M., Sidorkin, A. M., Costello, E., Watson, S., Mulligan, D., Honeychurch, S., Hodges, C. B., Sharples, M., Swindell, A., Frumin, I., Tlili, A., Slagter van Tryon, P. J., Bond, M., Bali, M., Leng, J., Zhang, K., Cukurova, M., Chiu, T. K. F., Lee, K., Hrastinski, S., Garcia, M. B., Sharma, R. C., Alexander, B., Zawacki-Richter, O., Huijser, H., Jandrić, P., Zheng, C., Shea, P., Duart, J. M., Themeli, C., Vorochkov, A., Sani-Bozkurt, S., Moore, R., & Asino, T. I. (2024). The Manifesto for Teaching and Learning in a Time of Generative AI: A Critical Collective Stance to Better Navigate the Future. *Open Praxis*, *16*(4), 487-513. https://doi.org/10.55982/openpraxis.16.4.777

Bringula, R. (2024). ChatGPT in a Programming Course: Benefits and Limitations. *Frontiers in Education*, *9*, 1-6. https://doi.org/10.3389/feduc.2024.1248705

Bucaioni, A., Ekedahl, H., Helander, V., & Nguyen, P. T. (2024). Programming with ChatGPT: How far can we go? *Machine Learning with Applications*, *15*, 1-10. https://doi.org/10.1016/j.mlwa.2024.100526

Budhiraja, R., Joshi, I., Challa, J. S., Akolekar, H. D., & Kumar, D. (2024). "It's not like Jarvis, but it's pretty close!" - Examining ChatGPT's Usage among Undergraduate Students in Computer Science. *Proceedings of the 26th Australasian Computing Education Conference*, 124-133. https://doi.org/10.1145/3636243.3636257

Chen, B., Lewis, C. M., West, M., & Zilles, C. (2024). Plagiarism in the Age of Generative AI: Cheating Method Change and Learning Loss in an Intro to CS Course. *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, 75-85. https://doi.org/10.1145/3657604.3662046

Chen, E., Huang, R., Chen, H.-S., Tseng, Y.-H., & Li, L.-Y. (2023). GPTutor: A ChatGPT-Powered Programming Tool for Code Explanation. *arXiv*. https://doi.org/10.48550/arXiv.2305.01863

Cheng, Y.-P., Lai, C.-F., Chen, Y.-T., Wang, W.-S., Huang, Y.-M., & Wu, T.-T. (2023). Enhancing Student's Computational Thinking Skills With Student-Generated Questions Strategy in a Game-based Learning Platform. *Computers & Education*, *200*, 1-20. https://doi.org/10.1016/j.compedu.2023.104794

Cong-Lem, N., Soyoof, A., & Tsering, D. (2024). A Systematic Review of the Limitations and Associated Opportunities of ChatGPT. *International Journal of Human–Computer Interaction*, 1-16. https://doi.org/10.1080/10447318.2024.2344142

Corney, M., Fitzgerald, S., Hanks, B., Lister, R., McCauley, R., & Murphy, L. (2014). 'Explain in Plain English' Questions Revisited: Data Structures Problems. *Proceedings of the 45th ACM technical symposium on Computer science education*, 591-596. https://doi.org/10.1145/2538862.2538911

Creswell, J. W. (2012). *Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research*. Pearson. https://books.google.com.ph/books?id=4PywcQAACAAJ

Dave, T., Athaluri, S. A., & Singh, S. (2023). ChatGPT in Medicine: An Overview of its Applications, Advantages, Limitations, Future Prospects, and Ethical Considerations. *Frontiers in Artificial Intelligence,*, *6*, 1-5. https://doi.org/10.3389/frai.2023.1169595

DePalma, K., Miminoshvili, I., Henselder, C., Moss, K., & AlOmar, E. A. (2024). Exploring ChatGPT's Code Refactoring Capabilities: An Empirical Study. *Expert Systems with Applications*, *249*, 1-26. https://doi.org/10.1016/j.eswa.2024.123602

Deriba, F. G., Sanusi, I. T., & Sunday, A. O. (2024). Enhancing Computer Programming Education using ChatGPT - A Mini Review. *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. https://doi.org/10.1145/3631802.3631848

Donthu, N., Kumar, S., Mukherjee, D., Pandey, N., & Lim, W. M. (2021). How to Conduct a Bibliometric Analysis: An Overview and Guidelines. *Journal of Business Research*, *133*, 285-296. https://doi.org/10.1016/j.jbusres.2021.04.070

Ellis, M. E., Casey, K. M., & Hill, G. (2024). ChatGPT and Python Programming Homework. *Decision Sciences Journal of Innovative Education*. https://doi.org/10.1111/dsji.12306

Feng, Y., Vanam, S., Cherukupally, M., Zheng, W., Qiu, M., & Chen, H. (2023). Investigating Code Generation Performance of ChatGPT with Crowdsourcing Social Data. *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 876-885. https://doi.org/10.1109/COMPSAC57700.2023.00117

Fiorella, L. (2023). Making Sense of Generative Learning. *Educational Psychology Review*, *35*(2), 1-42. https://doi.org/10.1007/s10648-023-09769-7

Garcia, M. B. (2021). Cooperative Learning in Computer Programming: A Quasi-Experimental Evaluation of Jigsaw Teaching Strategy with Novice Programmers. *Education and Information Technologies*, *26*(4), 4839-4856. https://doi.org/10.1007/s10639-021-10502-6

Garcia, M. B. (2023a). Facilitating Group Learning Using an Apprenticeship Model: Which Master is More Effective in Programming Instruction? *Journal of Educational Computing Research*, *61*(6), 1207-1231. https://doi.org/10.1177/07356331231170382

Garcia, M. B. (2023b). Fostering an Innovation Culture in the Education Sector: A Scoping Review and Bibliometric Analysis of Hackathons. *Innovative Higher Education*, *48*, 739-762. https://doi.org/10.1007/s10755-023-09651-y

Garcia, M. B. (2024). Profiling the Skill Mastery of Introductory Programming Students: A Cognitive Diagnostic Modeling Approach. *Education and Information Technologies*, 1-27. https://doi.org/10.1007/s10639-024-13039-6

Garcia, M. B., Enriquez, J. B., Adao, R., & Happonen, A. (2022). "Hey IDE, Display Hello World": Integrating a Voice Coding Approach in Hands-on Computer Programming Activities. *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. https://doi.org/10.1109/HNICEM57413.2022.10109412

Garcia, M. B., Revano Jr., T. F., Maaliw III, R. R., Lagrazon, P. G. G., Valderama, A. M. C., Happonen, A., Qureshi, B., & Yilmaz, R. (2023). Exploring Student Preference Between AI-Powered ChatGPT and Human-Curated Stack Overflow in Resolving Programming Problems and Queries. *2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 1-6. https://doi.org/10.1109/HNICEM60674.2023.10589162

Gasiba, T. E., Oguzhan, K., Kessba, I., Lechner, U., & Pinto-Albuquerque, M. (2023). I'm Sorry Dave, I'm Afraid I Can't Fix Your Code: On ChatGPT, CyberSecurity, and Secure Coding. *4th International Computer Programming Education Conference (ICPEC 2023)*, *112*, 1-12. https://doi.org/10.4230/OASIcs.ICPEC.2023.2

Groothuijsen, S., van den Beemt, A., Remmers, J. C., & van Meeuwen, L. W. (2024). AI Chatbots in Programming Education: Students' Use in a Scientific Computing Course and Consequences for Learning. *Computers and Education: Artificial Intelligence*, *7*, 1-9. https://doi.org/10.1016/j.caeai.2024.100290

Guo, P. (2013). Helping Scientists, Engineers to Work Up to 100 Times Faster. *Communications of the ACM*, *56*(10), 12–13. https://doi.org/10.1145/2507771.2507775

Gutierrez, A. D. C., Denny, P., & Luxton-Reilly, A. (2024). Evaluating Automatically Generated Contextualised Programming Exercises. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 289-295. https://doi.org/10.1145/3626252.3630863

Hartley, K., Hayak, M., & Ko, U. H. (2024). Artificial Intelligence Supporting Independent Student Learning: An Evaluative Case Study of ChatGPT and Learning to Code. *Education Sciences*, *14*(2), 1-12. https://doi.org/10.3390/educsci14020120

Husain, A. (2024). Potentials of ChatGPT in Computer Programming: Insights from Programming Instructors. *Journal of Information Technology Education: Research*, *23*, 1-25. https://doi.org/10.28945/5240

International Labour Organization. (2021). *Changing Demand for Skills in Digital Economies and Societies: Literature Review and Case Studies from Low- and Middle-Income Countries*. https://www.ilo.org/wcmsp5/groups/public/---ed_emp/---ifp_skills/documents/publication/wcms_831372.pdf

Jacques, L. (2023). Teaching CS-101 at the Dawn of ChatGPT. *ACM Inroads*, *14*(2), 40-46. https://doi.org/10.1145/3595634

Jarrett, K. (2022). The Utility of Game-Based Approaches within the PE Curriculum Design and Implementation Process to Develop "More Knowledgeable Others". *Strategies*, *35*(3), 3-10. https://doi.org/10.1080/08924562.2022.2052774

Joshi, I., Budhiraja, R., Dev, H., Kadia, J., Ataullah, M. O., Mitra, S., Akolekar, H. D., & Kumar, D. (2024). ChatGPT in the Classroom: An Analysis of Its Strengths and Weaknesses for Solving Undergraduate Computer Science Questions. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 625–631. https://doi.org/10.1145/3626252.3630803

Jošt, G., Taneski, V., & Karakatič, S. (2024). The Impact of Large Language Models on Programming Education and Student Learning Outcomes. *Applied Sciences*, *14*(10), 1-15. https://doi.org/10.3390/app14104115

Jukiewicz, M. (2024). The Future of Grading Programming Assignments in Education: The Role of ChatGPT in Automating the Assessment and Feedback Process. *Thinking Skills and Creativity*, *52*, 1-9. https://doi.org/10.1016/j.tsc.2024.101522

Kazemitabaar, M., Chow, J., Ma, C. K. T., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3544548.3580919

Khangura, S., Konnyu, K., Cushman, R., Grimshaw, J., & Moher, D. (2012). Evidence Summaries: The Evolution of a Rapid Review Approach. *Systematic Reviews*, *1*(1), 1-9. https://doi.org/10.1186/2046-4053-1-10

Kosar, T., Ostojić, D., Liu, Y. D., & Mernik, M. (2024). Computer Science Education in ChatGPT Era: Experiences from an Experiment in a Programming Course for Novice Programmers. *Mathematics*, *12*(5), 1-22. https://doi.org/10.3390/math12050629

Kucukkaya, A., Arikan, E., & Goktas, P. (2024). Unlocking ChatGPT's Potential and Challenges in Intensive Care Nursing Education and Practice: A Systematic Review With Narrative Synthesis. *Nursing Outlook*, *72*(6), 1-14. https://doi.org/10.1016/j.outlook.2024.102287

Kuramitsu, K., Obara, Y., Sato, M., & Obara, M. (2023). KOGI: A Seamless Integration of ChatGPT into Jupyter Environments for Programming Education. *Proceedings of the 2023 ACM SIGPLAN International Symposium on SPLASH-E*, 50–59. https://doi.org/10.1145/3622780.3623648

Lau, S., & Guo, P. (2023). From "Ban It Till We Understand It" to "Resistance is Futile": How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot. *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, *1*, 106-121. https://doi.org/10.1145/3568813.3600138

Lee, S., & Song, K.-S. (2024). Teachers' and Students' Perceptions of AI-generated Concept Explanations: Implications for Integrating Generative Ai in Computer Science Education. *Computers and Education: Artificial Intelligence*, *7*, 1-13. https://doi.org/10.1016/j.caeai.2024.100283

Leinonen, J., Hellas, A., Sarsa, S., Reeves, B., Denny, P., Prather, J., & Becker, B. A. (2023). Using Large Language Models to Enhance Programming Error Messages. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 563-569. https://doi.org/10.1145/3545945.3569770

Li, K., Hong, S., Fu, C., Zhang, Y., & Liu, M. (2023). Discriminating Human-authored from ChatGPT-Generated Code Via Discernable Feature Analysis. *2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 120-127. https://doi.org/10.1109/ISSREW60843.2023.00059

Liu, Y. (2023). Leveraging the Power of AI in Undergraduate Computer Science Education: Opportunities and Challenges. *2023 IEEE Frontiers in Education Conference (FIE)*, 1-5. https://doi.org/10.1109/FIE58773.2023.10343474

Maher, M., Tadimalla, S., & Dhamani, D. (2023). An Exploratory Study on the Impact of AI Tools on the Student Experience in Programming Courses: an Intersectional Analysis Approach. *2023 IEEE Frontiers in Education Conference (FIE)*, 1-5. https://doi.org/10.1109/FIE58773.2023.10343037

Mahon, J., Namee, B. M., & Becker, B. A. (2024). Guidelines for the Evolving Role of Generative AI in Introductory Programming Based on Emerging Practice. *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, 10–16. https://doi.org/10.1145/3649217.3653602

Marwan, S., Gao, G., Fisk, S., Price, T. W., & Barnes, T. (2020). Adaptive Immediate Feedback Can Improve Novice Programming Engagement and Intention to Persist in Computer Science. *Proceedings of the 2020 ACM Conference on International Computing Education Research*, 194-203. https://doi.org/10.1145/3372782.3406264

OECD. (2019). *Economic Outlook for Southeast Asia, China and India 2020*. https://doi.org/10.1787/1ba6cde0-en

Ouh, E. L., Gan, B. K. S., Shim, K. J., & Wlodkowski, S. (2023). ChatGPT, Can You Generate Solutions for my Coding Exercises? An Evaluation on its Effectiveness in an undergraduate Java Programming Course. *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, 54-60. https://doi.org/10.1145/3587102.3588794

Park, A., & Kim, T. (2025). Code Suggestions and Explanations in Programming Learning: Use of ChatGPT and Performance. *The International Journal of Management Education*, *23*(2), 1-17. https://doi.org/10.1016/j.ijme.2024.101119

Petrovska, O., Clift, L., Moller, F., & Pearsall, R. (2024). Incorporating Generative AI into Software Development Education. *Proceedings of the 8th Conference on Computing Education Practice*, 37-40. https://doi.org/10.1145/3633053.3633057

Piccolo, S. R., Denny, P., Luxton-Reilly, A., Payne, S. H., & Ridge, P. G. (2023). Evaluating a Large Language Model's Ability to Solve Programming Exercises from an Introductory Bioinformatics Course. *PLOS Computational Biology*, *19*(9), 1-16. https://doi.org/10.1371/journal.pcbi.1011511

Popovici, M.-D. (2023). ChatGPT in the Classroom. Exploring Its Potential and Limitations in a Functional Programming Course. *International Journal of Human–Computer Interaction*, 1-12. https://doi.org/10.1080/10447318.2023.2269006

Qureshi, B. (2023). ChatGPT in Computer Science Curriculum Assessment: An Analysis of Its Successes and Shortcomings. *Proceedings of the 2023 9th International Conference on e-Society, e-Learning and e-Technologies*, 7-13. https://doi.org/10.1145/3613944.3613946

Randall, N., Wäckerle, D., Stein, N., Goßler, D., & Bente, S. (2024). What an AI-Embracing Software Engineering Curriculum Should Look Like: An Empirical Study. *IEEE Software*, *41*(2), 36-43. https://doi.org/10.1109/MS.2023.3344682

Rose, K., Massey, V., Marshall, B., & Cardon, P. (2023). IS Professors' Perspectives on AI-Assisted Programming. *Issues in Information Systems*, *24*(2), 178-190. https://doi.org/10.48009/2_iis_2023_115

Savelka, J., Agarwal, A., An, M., Bogart, C., & Sakr, M. (2023). Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses. *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, *1*, 78-92. https://doi.org/10.1145/3568813.3600142

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2020). A Meta-Analysis of Teaching and Learning Computer Programming: Effective Instructional Approaches and Conditions. *Computers in Human Behavior*, *109*, 1-18. https://doi.org/10.1016/j.chb.2020.106349

Shaka, M., Carraro, D., & Brown, K. N. (2023). Personalised Programming Education with Knowledge Tracing. *Proceedings of the 2023 Conference on Human Centered Artificial Intelligence: Education and Practice*. https://doi.org/10.1145/3633083.3633220

Silva, C. A., Ramos, F. N., de Moraes, R. V., & Santos, E. L. (2024). ChatGPT: Challenges and Benefits in Software Programming for Higher Education. *Sustainability*, *16*(3), 1-23. https://doi.org/10.3390/su16031245

Spasić, A. J., & Janković, D. S. (2023). Using ChatGPT Standard Prompt Engineering Techniques in Lesson Preparation: Role, Instructions and Seed-Word Prompts. *2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, 47-50. https://doi.org/10.1109/ICEST58410.2023.10187269

Speth, S., Meißner, N., & Becker, S. (2023). Investigating the Use of AI-Generated Exercises for Beginner and Intermediate Programming Courses: A ChatGPT Case Study. *2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T)*, 142-146. https://doi.org/10.1109/CSEET58097.2023.00030

Stone, I. (2023). Exploring the Research Gap: Generative AI and Learning of Python Programming among Post-Primary Students. *Proceedings of the 2023 Conference on Human Centered Artificial Intelligence: Education and Practice*. https://doi.org/10.1145/3633083.3633099

MANUEL B. GARCIA
www.manuelgarcia.info

Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024). Would ChatGPT-Facilitated Programming Mode Impact College Students' Programming Behaviors, Performances, and Perceptions? An Empirical Study. *International Journal of Educational Technology in Higher Education*, *21*(1), 1-22. https://doi.org/10.1186/s41239-024-00446-5

Tricco, A. C., Antony, J., Zarin, W., Strifler, L., Ghassemi, M., Ivory, J., Perrier, L., Hutton, B., Moher, D., & Straus, S. E. (2015). A Scoping Review of Rapid Review Methods. *BMC Medicine*, *13*(1), 1-15. https://doi.org/10.1186/s12916-015-0465-6

Tsai, C.-Y. (2019). Improving Students' Understanding of Basic Programming Concepts Through Visual Programming Language: The Role of Self-Efficacy. *Computers in Human Behavior*, *95*, 224-232. https://doi.org/10.1016/j.chb.2018.11.038

Tuomi, P., Multisilta, J., Saarikoski, P., & Suominen, J. (2018). Coding Skills as a Success Factor for a Society. *Education and Information Technologies*, *23*(1), 419-434. https://doi.org/10.1007/s10639-017-9611-4

Valový, M., & Buchalcevova, A. (2023). The Psychological Effects of AI-Assisted Programming on Students and Professionals. *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 385-390. https://doi.org/10.1109/ICSME58846.2023.00050

Vieira, C., Magana, A. J., Falk, M. L., & Garcia, R. E. (2017). Writing In-Code Comments to Self-Explain in Computational Science and Engineering Education. *ACM Transactions on Computing Education*, *17*(4), 1-21. https://doi.org/10.1145/3058751

Wieser, M., Schöffmann, K., Stefanics, D., Bollin, A., & Pasterk, S. (2023). Investigating the Role of ChatGPT in Supporting Text-Based Programming Education for Students and Teachers. Informatics in Schools. Beyond Bits and Bytes: Nurturing Informatics Intelligence in Education, Cham.

Yan, D., Gao, Z., & Liu, Z. (2023). A Closer Look at Different Difficulty Levels Code Generation Abilities of ChatGPT. *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 1887-1898. https://doi.org/10.1109/ASE56229.2023.00096

Yang, A. C. M., Lin, J.-Y., Lin, C.-Y., & Ogata, H. (2024). Enhancing Python Learning With PyTutor: Efficacy of a ChatGPT-based Intelligent Tutoring System in Programming Education. *Computers and Education: Artificial Intelligence*, *7*, 1-15. https://doi.org/10.1016/j.caeai.2024.100309

Yilmaz, R., & Yilmaz, F. G. K. (2023). Augmented Intelligence in Programming Learning: Examining Student Views on the Use of ChatGPT for Programming Learning. *Computers in Human Behavior: Artificial Humans*, *1*(2), 1-7. https://doi.org/10.1016/j.chbah.2023.100005

York, E. (2023). Evaluating ChatGPT: Generative AI in UX Design and Web Development Pedagogy. *Proceedings of the 41st ACM International Conference on Design of Communication*, 197-201. https://doi.org/10.1145/3615335.3623035

Zheng, Y. (2023). ChatGPT for Teaching and Learning: An Experience from Data Science Education. *Proceedings of the 24th Annual Conference on Information Technology Education*, 66–72. https://doi.org/10.1145/3585059.3611431

MANUEL B. GARCIA
www.manuelgarcia.info

RELATED RESEARCH

Research Article

Profiling the Skill Mastery of Introductory Programming Students: A Cognitive Diagnostic Modeling Approach

Garcia, M. B. (2024). *Education and Information Technologies*, 1-27. https://doi.org/10.1007/s10639-024-13039-6.

Conference Paper

Exploring Student Preference Between AI-Powered ChatGPT and Human-Curated Stack Overflow in Resolving Programming Problems and Queries

Garcia, M. B., Revano Jr., T. F., Maaliw III, R. R., Lagrazon, P. G. G., Valderama, A. M. C., Happonen, A., Qureshi, B., & Yilmaz, R. (2023). *2023 IEEE 15th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 1-6. https://doi.org/10.1109/HNICEM60674.2023.10589162.

Conference Paper

Facilitating Group Learning Using an Apprenticeship Model: Which Master is More Effective in Programming Instruction?

Garcia, M. B. (2023). *Journal of Educational Computing Research*, 61(6), 1207-1231. https://doi.org/10.1177/07356331231170382.

LET'S COLLABORATE!

If you are looking for research collaborators, please do not hesitate to contact me at mbgarcia@feutech.edu.ph.

ABOUT THE CORRESPONDING AUTHOR:

Manuel B. Garcia is a professor of information technology and the founding director of the Educational Innovation and Technology Hub (EdITH) at FEU Institute of Technology, Manila, Philippines. His interdisciplinary research interest includes topics that, individually or collectively, cover the disciplines of education and information technology. He is a licensed professional teacher and a proud member of the National Research Council of the Philippines – an attached agency to the country's Department of Science and Technology (DOST-NRCP).